

A FEATURE SELECTION METHOD FOR MULTIVARIATE PERFORMANCE MEASURES

QI MAO AND IVOR W. TSANG

ABSTRACT. Feature selection with specific multivariate performance measures is the key to the success of many applications such as information retrieval. In this paper, we propose a feature selection method for multivariate performance measures. The proposed method forms an optimization problem with exponential size of both feature groups and label configurations for a given dataset. To address this problem, a two-layer cutting plane algorithm is proposed. The outer layer performs group feature generation; while the inner layer learns the label configuration for multivariate performance measures. Comprehensive experiments on large-scale and high-dimensional real world datasets show that the proposed method can significantly outperform l_1 -SVM and SVM-RFE when choosing a small subset of features, and achieve significantly improved performances over SVM^{perf} in terms of F_1 -score. It also learns a sparse yet effective decision rule for multivariate performance measures.

1. INTRODUCTION

Feature selection is crucial to many applications such as text mining, image retrieval and bioinformatics. These applications usually contain a huge amount of features which incur very high computational costs for analysis. Pruning non-informative or noisy features can usually improve the generalization performance. Moreover, a small set of feature is beneficial to visualize or interpret the results. One of the most widely used criteria for feature selection is the maximum margin criterion, particularly on Support Vector Machine (SVM). The weights of the SVM model can be used for feature selection in two directions. One way is to consider the sparsity of weights by replacing l_2 -norm regularization with l_1 -norm [28, 5, 15]. Recently, Yuan et al. [24] conducted a thorough study to compare several recently developed l_1 -regularized algorithms. From their study, coordinate descent method using one-dimensional Newton direction (CDN) can achieve the state-of-the-art performance on solving l_1 -regularized problems.

To achieve a sparser solution, the Approximation of the zero norm Minimization (AROM) was proposed [21]. Its resultant problem is non-convex, so it suffers from local optima. Despite this, the recent results [12] and theoretical studies [26] also showed that l_p models ($p < 1$) even with a local optimal solution achieves better parameter estimation performances than convex l_1 models, which are asymptotically biased [12]. Chan et al. [3] also proposed two convex relaxations to l_0 -SVM, but they are computationally expensive, especially for high dimensional datasets. Another way is to sort the weights and remove the smallest weights iteratively in SVM-Recursive Feature Elimination (SVM-RFE)[6]. However, as discussed in [23], such nested “monotonic” feature selection scheme leads to the suboptimal performance. Non-monotonic feature selection (NMMKL) [23] was proposed to solve this problem, but each feature corresponding to one kernel makes NMMKL

Qi Mao and Ivor W. Tsang are with School of Computer Engineering, Nanyang Technological University, Singapore 639798, e-mail {QMAO1,IvorTsang}@ntu.edu.sg.

infeasible for high dimensional problems. Recently, Tan et al. [18] proposed Feature Generating Machine(FGM), which shows great scalability to non-monotonic feature selection on large-scale and very high-dimensional datasets. However, since FGM is formulated for the 0-1 loss function, it is not appropriate for other specific applications.

Depending on applications, specific performance measures are usually required to evaluate the success of a learning algorithm. In text classification, for example, F_1 -score and Precision/Recall Breakeven Point(PRBEP) are used to evaluate classifier performance; while error rate is not suitable due to a large imbalance between positive and negative examples [8]. Thereafter, SVM^{perf} [8] was proposed for multivariate performance measures. As shown in [8], optimizing the learning model subject to the specific multivariate performance measures can boost the corresponding performance. However, for high dimension data, such as image and document retrieval, it is urgent to perform feature selection since the noisy or non-informative features may degrade performance measures. Moreover, feature selection helps significantly speed up the prediction on high dimensional data in real world retrieval applications.

In this paper, we propose to make FGM suitable for multivariate performance measures. By transplanting a 0-1 control variable associated with each feature into multivariate prediction framework [8], we can derive the modified FGM for multivariate performance measures, namely FGM^{perf} . Note the resultant optimization problem is more complicated than that of FGM due to the exponential size of both the subset of features and label configuration for all examples. Under this situation, existing Multiple Kernel Learning (MKL) algorithms are infeasible to be utilized for solving the MKL problem inside FGM because of the exponential number of constraints in the primal form or optimization variables in the dual form.

To this end, we propose a two-layer cutting plane algorithm: the outer layer performs group feature generation; while the inner one selects label configurations for multivariate performance measures. Comprehensive experiments on several large-scale and very high-dimensional real world datasets show that the proposed method yields comparable performance with state-of-the-art feature selection methods on the 0-1 loss, and outperforms SVM^{perf} using all features in term of multivariate performance measures.

In the rest of this paper, we denote the transpose of a vector/matrix by the superscript T and l_p norm of a vector \mathbf{v} by $\|\mathbf{v}\|_p$. Binary operator \odot represents the elementwise product between two vectors/matrices.

2. MULTIVARIATE PERFORMANCE MEASURES

A large class of multivariate performance measures, such as F_1 -score, Recall@ k and Precision/Recall Breakeven Point(PRBEP) are non-linear and multivariate. Their decision theoretic risks cannot be decomposed into expectations over individual examples, and are difficult to be optimized directly. [8] proposed to formulate the learning problem as a multivariate prediction of all examples in the dataset in order to accommodate this problem based on sparse approximation algorithm for structural SVMs [20]. Given a training sample of input-output pairs $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n) \in \mathcal{X} \times \mathcal{Y}$ drawn from some fixed but unknown probability distribution with $\mathcal{X} \in R^m$ and $\mathcal{Y} \in \{-1, +1\}$. The learning problem is treated as a multivariate prediction problem by defining the hypotheses \bar{h} that map a tuple $\bar{\mathbf{x}} \in \bar{\mathcal{X}}$ of n feature vectors $\bar{\mathbf{x}} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ to a tuple $\bar{y} \in \bar{\mathcal{Y}}$ of n labels $\bar{y} = (y_1, \dots, y_n)$, $\bar{h} : \bar{\mathcal{X}} \rightarrow \bar{\mathcal{Y}}$ where $\bar{\mathcal{X}} = \mathcal{X} \times \dots \times \mathcal{X}$ and $\bar{\mathcal{Y}} \subseteq \{-1, +1\}^n$ is the set of all admissible label

vectors. The linear discriminant function are then defined as follows,

$$(1) \quad \bar{h}_{\mathbf{w}}(\bar{\mathbf{x}}) = \arg \max_{\bar{y}' \in \bar{\mathcal{Y}}} \left\{ \sum_{i=1}^n y'_i \mathbf{w}^T \mathbf{x}_i \right\},$$

where \mathbf{w} is the weight vector. The multivariate loss functions can be easily incorporated into structural SVM in one slack variable formula as follows,

$$(2) \quad \min_{\mathbf{w}, \xi \geq 0} \quad \frac{1}{2} \|\mathbf{w}\|^2 + C\xi$$

$$\text{s.t.} \quad \forall \bar{y}' \in \bar{\mathcal{Y}} \setminus \bar{y} : \mathbf{w}^T \sum_{i=1}^n (y_i - y'_i) \mathbf{x}_i \geq \Delta(\bar{y}, \bar{y}') - \xi,$$

where $\Delta(\bar{y}, \bar{y}')$ is some type of multivariate loss functions. This optimization problem is a convex optimization problem, but there is the exponential size of $\bar{\mathcal{Y}}$. However, this problem can be solved in polynomial time by adopting the sparse approximation algorithm of structural SVM.

3. FEATURE GENERATING MACHINE (FGM)

Feature Generating Machine(FGM) was proposed by [18] to learn a sparse solution to SVM. The discriminant function of this sparse SVM model is represented as follows,

$$(3) \quad h_{\mathbf{w}}(\mathbf{x}) = \text{sign}((\mathbf{w} \odot \mathbf{d})^T \mathbf{x}),$$

where $h_{\mathbf{w}} : \mathcal{X} \rightarrow \mathcal{Y}$, and $\mathbf{d} = [d_1, \dots, d_m]^T$ is a vector of 0-1 control variables in the domain of $\mathcal{D} = \{\mathbf{d} \mid \sum_{j=1}^m d_j \leq B, d_j \in \{0, 1\}, \forall j = 1, \dots, m\}$. The parameter B is a budget to control the sparsity of \mathbf{d} . For clarity, we call a feature configuration $\mathbf{d}^t \in \mathcal{D}$ as a group. Namely, if the i th feature is selected into the group \mathbf{d}^t , then $d_i^t = 1$, otherwise 0. The sparse representation of a group is a set of indices with $d_i^t = 1$. Then, FGM attempts to learn this discriminant function and performs feature selection simultaneously by solving the following optimization problem,

$$(4) \quad \min_{\mathbf{d} \in \mathcal{D}} \min_{\mathbf{w}, \xi, \rho} \quad \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{2} \sum_{i=1}^n \xi_i^2 - \rho$$

$$\text{s.t.} \quad y_i \mathbf{w}^T (\mathbf{x}_i \odot \mathbf{d}) \geq \rho - \xi_i, \forall i = 1, \dots, n,$$

where $\rho / \|\mathbf{w}\|$ is the margin separation. This problem is in form of a mixed integer programming problem, which is computationally expensive due to the exponential size of \mathcal{D} . [18] proposed to solve this problem by a cutting plane algorithm which generates a pool of the most violated feature subsets and then combines them via MKL algorithm iteratively.

4. FEATURE SELECTION FOR MULTIVARIATE PERFORMANCE MEASURES

In this Section, we illustrate the proposed method in details. By a simple combination of the discriminant functions in (1) and (3), we can obtain a new discriminant functions $\tilde{h}_{\mathbf{w}}(\mathbf{x})$ as

$$(5) \quad \tilde{h}_{\mathbf{w}}(\mathbf{x}) = \arg \max_{\bar{y}' \in \bar{\mathcal{Y}}} \left\{ \sum_{i=1}^n y'_i (\mathbf{w} \odot \mathbf{d})^T \mathbf{x}_i \right\}.$$

To optimize the multivariate loss functions and learn a sparse feature representation, we propose to solve the following problem,

$$(6) \quad \min_{\mathbf{d} \in \mathcal{D}} \min_{\mathbf{w}, \xi \geq 0} \frac{1}{2} \|\mathbf{w}\|^2 + C\xi$$

$$\text{s.t. } \forall \bar{\mathbf{y}}' \in \bar{\mathcal{Y}} \setminus \bar{\mathbf{y}} : \mathbf{w}^T \frac{1}{n} \sum_{i=1}^n (y_i - y'_i) (\mathbf{x}_i \odot \mathbf{d}) \geq \tilde{\Delta}(\bar{\mathbf{y}}, \bar{\mathbf{y}}') - \xi,$$

where $\tilde{\Delta} = \frac{1}{n} \Delta(\bar{\mathbf{y}}, \bar{\mathbf{y}}')$ is the average of multivariate loss. This problem turns out to be even more complicated to be solved due to the exponential size of both \mathcal{D} and $\bar{\mathcal{Y}}$. To this end, we propose a two-layer cutting plane algorithm to solve it efficiently and effectively. The two layers, group feature generation and group feature selection, will be described in Section 4.1 and 4.2, respectively. The two-layer cutting plane algorithm will be presented in Section 4.3 and 4.4.

4.1. Group Feature Generation. This layer is similar to FGM [18] to generate a pool of the most violated feature subsets, but the dual form of Problem (6) has exponential number of dual variables. The partial dual with respect to \mathbf{w}, ξ can be obtained as follows,

$$\max_{\alpha \in \mathcal{A}} -\frac{1}{2} \sum_{\bar{\mathbf{y}}'} \sum_{\bar{\mathbf{y}}''} \alpha_{\bar{\mathbf{y}}'} \alpha_{\bar{\mathbf{y}}''} Q_{\bar{\mathbf{y}}', \bar{\mathbf{y}}''}^{\mathbf{d}} + \sum_{\bar{\mathbf{y}}'} \alpha_{\bar{\mathbf{y}}'} b_{\bar{\mathbf{y}}'},$$

where α is the dual variables, $Q_{\bar{\mathbf{y}}', \bar{\mathbf{y}}''}^{\mathbf{d}} = \langle \mathbf{a}_{\bar{\mathbf{y}}'}, \mathbf{a}_{\bar{\mathbf{y}}''} \rangle$, $\mathbf{a}_{\bar{\mathbf{y}}'} = \frac{1}{n} \sum_{i=1}^n (y_i - y'_i) (\mathbf{x}_i \odot \mathbf{d})$, $b_{\bar{\mathbf{y}}'} = \frac{1}{n} \Delta(\bar{\mathbf{y}}, \bar{\mathbf{y}}')$, and $\mathcal{A} = \{\sum_{\bar{\mathbf{y}}'} \alpha_{\bar{\mathbf{y}}'} \leq C, \alpha \geq 0\}$. By denoting $S(\alpha, \mathbf{d}) = -\frac{1}{2} \sum_{\bar{\mathbf{y}}'} \sum_{\bar{\mathbf{y}}''} \alpha_{\bar{\mathbf{y}}'} \alpha_{\bar{\mathbf{y}}''} Q_{\bar{\mathbf{y}}', \bar{\mathbf{y}}''}^{\mathbf{d}} + \sum_{\bar{\mathbf{y}}'} \alpha_{\bar{\mathbf{y}}'} b_{\bar{\mathbf{y}}'}$, Problem (6) turns out to be $\min_{\mathbf{d} \in \mathcal{D}} \max_{\alpha \in \mathcal{A}} S(\alpha, \mathbf{d})$.

Following [11, 18], we introduce a mild convex relaxation for above problem. According to the minimax inequality [10], we can interchange the *min* and *max* to obtain a lower-bounded problem, $\max_{\alpha \in \mathcal{A}} \min_{\mathbf{d} \in \mathcal{D}} S(\alpha, \mathbf{d})$. We further denote $\mathcal{F}_{\mathbf{d}}(\alpha) = -S(\alpha, \mathbf{d})$, then we have

$$(7) \quad \min_{\alpha \in \mathcal{A}} \max_{\mathbf{d} \in \mathcal{D}} \mathcal{F}_{\mathbf{d}}(\alpha) \quad \text{or} \quad \min_{\alpha \in \mathcal{A}, \gamma} \gamma : \gamma \geq \mathcal{F}_{\mathbf{d}}(\alpha), \forall \mathbf{d} \in \mathcal{D}.$$

Though there are exponential number of \mathbf{d} 's in \mathcal{D} , fortunately, only a few constraints in (7) are active at the optimality, and including only a subset of these constraints can usually lead to a very tight approximation of the original optimization problem. Cutting plane algorithm [9] could be used here to solve this problem. Since $\max_{\mathbf{d} \in \mathcal{D}} \mathcal{F}_{\mathbf{d}}(\alpha) \geq \mathcal{F}_{\mathbf{d}^t}(\alpha), \forall \mathbf{d}^t \in \mathcal{D}$, the lower bound approximation of (7) can be obtained by $\max_{\mathbf{d} \in \mathcal{D}} \mathcal{F}_{\mathbf{d}}(\alpha) \geq \max_{t=1, \dots, T} \mathcal{F}_{\mathbf{d}^t}(\alpha)$. Then we can minimize the lower bound of (7) by,

$$(8) \quad \min_{\alpha \in \mathcal{A}} \max_{t=1, \dots, T} \mathcal{F}_{\mathbf{d}^t}(\alpha) \quad \text{or} \quad \min_{\alpha \in \mathcal{A}, \gamma} \gamma : \gamma \geq \mathcal{F}_{\mathbf{d}^t}(\alpha), \forall t = 1, \dots, T.$$

As from [14], such cutting plane algorithm can converge to a robust optimal solution within tens of iterations with the exact worst-case analysis. Specifically, for a fixed α^t , the worst-case analysis can be done by solving,

$$(9) \quad \mathbf{d}^t = \arg \max_{\mathbf{d} \in \mathcal{D}} \mathcal{F}_{\mathbf{d}}(\alpha^t),$$

which is referred to as the group generation procedure. However, Problem (8) and (9) cannot be solved directly due to the exponential size of α where each entry of α corresponds to one configuration in $\bar{\mathcal{Y}}$.

4.2. Group Feature Selection. By introducing dual variables $\mu = [\mu_1, \mu_2, \dots, \mu_T]^T \geq 0$, we can transform (8) to a MKL problem as follows,

$$(10) \quad \max_{\alpha \in \mathcal{A}} \min_{\mu \in \mathcal{M}_T} -\frac{1}{2} \sum_{\bar{y}'} \sum_{\bar{y}''} \alpha_{\bar{y}'} \alpha_{\bar{y}''} \left(\sum_{t=1}^T \mu_t Q_{\bar{y}', \bar{y}''}^{\mathbf{d}^t} \right) + \sum_{\bar{y}'} \alpha_{\bar{y}'} b_{\bar{y}'},$$

where $\mathcal{M}_T = \{\sum_{t=1}^T \mu_t = 1, \mu \geq 0\}$.

However, due to the exponential size of α , the complexity of Problem (10) remains. In this case, state-of-the-art multiple kernel learning algorithms [17, 16, 22] do not work any more. The following proposition shows that we can indirectly solve Problem (10) in the primal form.

Proposition 1. *The primal form of Problem (10) is*

$$(11) \quad \min_{\mathbf{w}_1, \dots, \mathbf{w}_T, \xi \geq 0} \quad \frac{1}{2} \left(\sum_{t=1}^T \|\mathbf{w}_t\|_2 \right)^2 + C\xi$$

$$s.t. \quad \xi \geq b_{\bar{y}'} - \sum_{t=1}^T \langle \mathbf{w}_t, \mathbf{a}_{\bar{y}'}^t \rangle, \forall \bar{y}' \in \bar{\mathcal{Y}} \setminus \bar{y}.$$

According to KKT conditions, the solution of (11) is

$$(12) \quad \mathbf{w}_t = \mu_t \sum_{\bar{y}'} \alpha_{\bar{y}'} \mathbf{a}_{\bar{y}'}^t$$

where μ_t is a dual value of the t^{th} constraint of (8).

Here, we define the regularization term as $\Omega(\bar{\mathbf{w}}) = \frac{1}{2} \left(\sum_{t=1}^T \|\mathbf{w}_t\|_2 \right)^2$ with $\bar{\mathbf{w}} = [\mathbf{w}_1, \dots, \mathbf{w}_T]^T$ and the empirical risk function as

$$(13) \quad R_{emp}(\bar{\mathbf{w}}) = \frac{1}{n} \max \left(0, \max_{\bar{y}' \in \bar{\mathcal{Y}} \setminus \bar{y}} b_{\bar{y}'} - \sum_{t=1}^T \langle \mathbf{w}_t, \mathbf{a}_{\bar{y}'}^t \rangle \right),$$

which is a convex but non-smooth function w.r.t $\bar{\mathbf{w}}$. Then we can apply the bundle method [19] to solve this primal problem. Problem (11) is transformed as

$$\min_{\bar{\mathbf{w}}} \mathcal{J}(\bar{\mathbf{w}}) = \Omega(\bar{\mathbf{w}}) + C R_{emp}(\bar{\mathbf{w}}).$$

Since $R_{emp}(\bar{\mathbf{w}})$ is a convex function, its subgradient exists everywhere in its domain [7]. Suppose $\bar{\mathbf{w}}^k$ is a point in where $R_{emp}(\bar{\mathbf{w}})$ is finite, we can formulate the lower bound according to the definition of subgradient,

$$R_{emp}(\bar{\mathbf{w}}) \geq R_{emp}(\bar{\mathbf{w}}^k) + \langle \bar{\mathbf{w}} - \bar{\mathbf{w}}^k, \mathbf{p}^k \rangle,$$

where subgradient $\mathbf{p}^k \in \partial_{\bar{\mathbf{w}}} R_{emp}(\bar{\mathbf{w}}^k)$ is at $\bar{\mathbf{w}}^k$. Given subgradient sequence $\mathbf{p}^1, \mathbf{p}^2, \dots, \mathbf{p}^K$, the tighter lower bound for $R_{emp}(\bar{\mathbf{w}})$ can be stated as follows,

$$R_{emp}(\bar{\mathbf{w}}) \geq R_{emp}^K(\bar{\mathbf{w}}) = \max \left(0, \max_{1 \leq k \leq K} \langle \bar{\mathbf{w}}, \mathbf{p}^k \rangle + q^k \right),$$

where $q^k = R_{emp}(\bar{\mathbf{w}}^k) - \langle \bar{\mathbf{w}}^k, \mathbf{p}^k \rangle$. Following the bundle method [19], the criterion for selecting the next point $\bar{\mathbf{w}}^{K+1}$ is to solve the following optimization problem,

$$(14) \quad \begin{aligned} \min_{\mathbf{w}_1, \dots, \mathbf{w}_T, \xi \geq 0} \quad & \frac{1}{2} \left(\sum_{t=1}^T \|\mathbf{w}_t\|_2 \right)^2 + C\xi \\ \text{s.t.} \quad & \xi \geq \langle \bar{\mathbf{w}}, \mathbf{p}^k \rangle + q^k, \forall k = 1, \dots, K. \end{aligned}$$

The following Corollary shows that Problem (14) can be easily solved by QCQP solvers, and the number of variables is independent of the number of examples.

Corollary 1. *In terms of Proposition 1, the dual form of Problem (14) is*

$$(15) \quad \begin{aligned} \max_{\alpha \in \mathcal{A}_K} \max_{\theta} \quad & -\theta + \sum_{k=1}^K \alpha_k q^k \\ \text{s.t.} \quad & \frac{1}{2} \left\| \sum_{k=1}^K \alpha_k \mathbf{p}_t^k \right\|_2^2 \leq \theta, \forall t = 1, \dots, T, \end{aligned}$$

where $\mathcal{A}_K = \{\sum_{k=1}^K \alpha_k \leq C, \alpha_k \geq 0, \forall k = 1, \dots, K\}$, and which is a QCQP problem with $T + 1$ constraints and $K + 1$ variables.

Remark that Problem (14) is similar to the Support Kernel Machine (SKM) [2] in which the multiple Gaussian kernels are built on random subsets of features, with varying widths. However, our method can automatically choose the most violated subset of features as a group instead of a subset of random features. Such random features lead to a local optimum; while our method could guarantee the ϵ -optimality stated in Theorem 1. However, due to the combinational structure of \mathcal{D} in Problem (9), the current model can only work for linear kernel with different subsets of features. By setting $\mathcal{J}_K(\bar{\mathbf{w}}) = \Omega(\bar{\mathbf{w}}) + CR_{emp}^K(\bar{\mathbf{w}})$, the ϵ -optimal condition in Algorithm 1 is $\min_{0 \leq k \leq K} \mathcal{J}(\bar{\mathbf{w}}^k) - \mathcal{J}_K(\bar{\mathbf{w}}^K) \leq \epsilon$.

Algorithm 1 group_feature_selection

- 1: Input: $\bar{\mathbf{x}} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$, $\bar{\mathbf{y}} = (y_1, \dots, y_n)$, an initial group set \mathcal{W}
 - 2: $\bar{\mathcal{Y}} = \emptyset$, $k = 0$
 - 3: **repeat**
 - 4: $k = k + 1$
 - 5: Finding the most violated $\bar{\mathbf{y}}'$
 - 6: Compute \mathbf{p}^k and q^k
 - 7: $\bar{\mathcal{Y}} = \bar{\mathcal{Y}} \cup \{\bar{\mathbf{y}}'\}$
 - 8: Solving Problem (15) over \mathcal{W} and $\bar{\mathcal{Y}}$
 - 9: **until** ϵ -optimal
-

4.3. Two-Layer Cutting Plane Algorithm. Algorithm 1 can obtain the ϵ -optimal solution for the original dual problem (8). By denoting $\mathcal{G}_d(\alpha) = \frac{1}{2} \left\| \sum_{k=1}^K \alpha_k \mathbf{p}_t^k \right\|_2^2 - \sum_{k=1}^K \alpha_k q^k$, the group feature generation layer can directly use the ϵ -optimal solution of the objective $\mathcal{G}_d(\alpha)$ to approximate the original objective $\mathcal{F}_d(\alpha)$. The two-layer cutting plane algorithm is presented in Algorithm 2. From the description of Algorithm 2, it is clear to see that groups are dynamically generated and augmented into active set \mathcal{W} for group selection.

Algorithm 2 The Two-Layer Method

```

1: Input:  $\bar{\mathbf{x}} = (\mathbf{x}_1, \dots, \mathbf{x}_n), \bar{\mathbf{y}} = (y_1, \dots, y_n)$ 
2:  $\mathcal{W} = \emptyset, t = 0$ 
3: repeat
4:    $t = t + 1$ 
5:   Finding the most violated  $\mathbf{d}^t$ 
6:    $\mathcal{W} = \mathcal{W} \cup \{\mathbf{d}^t\}$ 
7:   group_feature_selection( $\bar{\mathbf{x}}, \bar{\mathbf{y}}, \mathcal{W}$ )
8: until Convergence

```

In terms of the convergence proof of FGM in [18], we can obtain the following theorem to illustrate the approximation with an ϵ -optimal solution to the original problem.

Theorem 1. *After Algorithm 2 stops in a finite number of steps, the difference between optimal solution (\mathbf{d}^*, α^*) of Problem (10) and the solution (\mathbf{d}, α) of Algorithm 2 is $S(\alpha^*, \mathbf{d}^*) - \Theta_{\mathbf{d}}(\alpha) \leq \epsilon$.*

4.4. Finding the Most Violated $\bar{\mathbf{y}}'$ and \mathbf{d} . Algorithm 1 and Algorithm 2 need to find the most violated $\bar{\mathbf{y}}'$ and \mathbf{d} , respectively. In this subsection, we discuss how to obtain these quantities efficiently.

Algorithm 1 needs to calculate the subgradient of the empirical risk function $R_{emp}^K(\bar{\mathbf{w}})$. Since $R_{emp}^K(\bar{\mathbf{w}})$ is a pointwise supremum function, the subgradient should be in the convex hull of the gradient of the decomposed functions with the largest objective. Here, we just take one of this subgradient by solving

$$(16) \quad \bar{\mathbf{y}}^k = \arg \max_{\bar{\mathbf{y}}' \in \mathcal{Y} \setminus \bar{\mathbf{y}}} \Delta(\bar{\mathbf{y}}', \bar{\mathbf{y}}) - \sum_{i=1}^n (y_i - y'_i) v_i,$$

where $v_i = \sum_{t=1}^T \mathbf{w}_t^T (\mathbf{x}_i \odot \mathbf{d}^t)$. After obtaining $\bar{\mathbf{y}}^k$, it is easy to compute $\mathbf{p}_t^k = -\frac{1}{n} \sum_{i=1}^n (y_i - y_i^k) (\mathbf{x}_i \odot \mathbf{d}^t)$ and $q^k = \frac{1}{n} \sum_{i=1}^n \Delta(\bar{\mathbf{y}}^k, \bar{\mathbf{y}})$.

For finding the most violated $\bar{\mathbf{y}}'$, it depends on how to define the loss $\Delta(\bar{\mathbf{y}}, \bar{\mathbf{y}}')$ in Problem (16). One of the instances is the hamming loss which can be decomposed and computed independently, $\Delta(\bar{\mathbf{y}}, \bar{\mathbf{y}}') = \sum_{i=1}^n \delta(y_i, y'_i)$, where δ is an indicator function with $\delta(y_i, y'_i) = 0$ if $y_i = y'_i$, otherwise 1. However, there are some multivariate performance measures which could not be solved independently. Fortunately, there are a series of structured loss functions, such as Area Under ROC (AUC), Average Precision (AP), ranking and contingency table scores and other measures listed in [8, 25, 19], which can be implemented efficiently in our algorithms. In this paper, we only use several multivariate performance measures based on contingency table as the showcases and their finding $\bar{\mathbf{y}}^k$ could be solved in time complexity $O(n^2)$ [8].

Given the true labels \mathbf{y} and predicted labels \mathbf{y}' , the contingency tables is defined as follows

	y=1	y=-1
y'=1	a	b
y'=-1	c	d

F_1 -score: The F_β -score is a weighted harmonic average of Precision and Recall. According to the contingency table, we can obtain

$$F_\beta = \frac{(1 + \beta^2)a}{(1 + \beta^2)a + b + \beta^2 c}.$$

TABLE 1. Datasets used in our experiments

Dataset	#classes	#features	#train. points	#test points
News20.binary	2	1,355,191	11,997	7,999
URL1	2	3,231,961	20,000	20,000
Image	5	10,800	1,200	800
Rcv1	53	47,236	15,564	518,571
Sector	105	55,197	6,412	3,207
News20	20	62,061	15,935	3,993

The most common choice is $\beta = 1$. The corresponding balanced F_1 measure loss can be written as $\Delta(a, b, c, d) = 100(1 - F_\beta)$. Then, Algorithm 2 in [8] can be directly applied.

Precision/Recall@k In search engine systems, most users scan only the first few links that are presented. In this situation, $Prec@k$ and $Rec@k$ measure the precision and recall of a classifier that predicts exactly k documents, i.e.,

$$Prec@k = \frac{a}{a+b}, \quad Rec@k = \frac{a}{a+c}$$

subject to $a + b = k$. The corresponding loss could be defined as $\Delta_{Prec@k} = 100(1 - Prec@k)$, $\Delta_{Rec@k} = 100(1 - Rec@k)$. And the procedure of finding most violated \mathbf{y} is similar to F-score, while the only difference is keeping constraint $a + b = k$ and removing $a + b \neq k$. In the evaluation part, we label all the k largest decision value as 1, and then calculate the values of $Prec@k$ and $Rec@k$.

Precision/Recall Break-Even Point The Precision/Recall Break-Even Point makes a precision and recall are equal. According to above definition, we can see PRBEP only adds a constraint $a + b = a + c$, or $b = c$. The corresponding loss could be defined as $\Delta_{PRBEP} = 100(1 - PRBEP)$. Finding the most violated \mathbf{y} should enforce the constraint $b = c$.

Now, we can simplify α in Problem (9) from the exponential size to T . Then finding the most violated \mathbf{d} in Algorithm 2 becomes

$$\begin{aligned}
(17) \quad \mathbf{d}^t &= \arg \max_{\mathbf{d} \in \mathcal{D}} \mathcal{G}_{\mathbf{d}}(\alpha^t) \\
&= \arg \max_{\mathbf{d} \in \mathcal{D}} \frac{1}{2} \left\| \frac{1}{n} \sum_{k=1}^K \alpha_k^t \sum_{i=1}^n (y_i - y_i^k) (\mathbf{x}_i \odot \mathbf{d}) \right\|^2 \\
&= \arg \max_{\mathbf{d} \in \mathcal{D}} \frac{1}{2n^2} \sum_{j=1}^D c_j^2 d_j
\end{aligned}$$

where $c_j = \sum_{k=1}^K \alpha_k^t \sum_{i=1}^n (y_i - y_i^k) \mathbf{x}_{i,j}$. With the budget constraint $\sum_{i=1}^m d_i \leq B$ in \mathcal{D} , (17) can be solved by first sorting c_j^2 's in the descent order and then setting the first B numbers corresponding to d_j^t to 1 and the rest to 0. This takes only $O(m \log m)$ operations.

5. EXPERIMENTS

In this Section, we conduct extensive experiments to evaluate the performance of our proposed method and state-of-the-art feature selection methods: 1) SVM-RFE [6]; 2) l_1 -SVM; 3) FGM [18]. SVM-RFE and FGM use Liblinear software¹ as the QP solver for

¹<http://www.csie.ntu.edu.tw/~cjlin/liblinear/>

their SVM subproblems. For l_1 -SVM, we also use Liblinear software, which implements the state-of-the-art l_1 -SVM algorithm [24].

For convenience, we name our proposed two-layer cutting plane algorithm $\text{FGM}_{multi}^\Delta$, where Δ represents different type of multivariate performance measures. We implemented Algorithm 2 in MATLAB for all the multivariate performance measures listed above, using Mosek [1] as the QCQP solver for Problem (15) which yields a worse-case complexity of $O(KT^2)$. Since the values of both K and T are much smaller than the number of examples n and its dimensionality m , the QCQP is very efficient as well as more accurate for large-scale and high-dimensional datasets. Furthermore, the codes simultaneously solve the primal and its dual form. So the optimal μ and α can be obtained after solving Problem (15).

For a test pattern \mathbf{x} , the discriminant function can be obtained by $f(\mathbf{x}) = \langle \mathbf{w} \odot \tilde{\mathbf{d}}, \mathbf{x} \rangle$ where $\mathbf{w} = \sum_{i=1}^n \beta_i \mathbf{x}_i$, $\beta_i = \frac{1}{n} \sum_{k=1}^K \alpha_k (y_i - y_i^k)$, and $\tilde{\mathbf{d}} = \sum_{t=1}^T \mu_t \mathbf{d}^t$. This leads to the faster prediction since only a few of the selected features are involved. After computing \mathbf{p}^k , the matrices of Problem (15) can be incrementally updated, so it can be done totally in $O(TK^2)$.

5.1. Feature Selection for Accuracy. Since [8] has proven that $\text{SVM}_{multi}^\Delta$ with hamming loss, namely $\Delta_{Err}(\bar{y}, \bar{y}') = 2(b + c)$, is the same as SVM. In this subsection, we first evaluate the accuracy performances of $\text{FGM}_{multi}^\Delta$ for hamming loss function, namely $\text{FGM}_{multi}^{hamming}$ as well as other state-of-the-art feature selection methods. We compare these methods on two binary datasets, *News20.binary*² and *URL1* in Table 1. These two datasets are used in [18]. We use the standard training and test sets to evaluate the prediction performance of feature selection methods in this experiment.

We test FGM and SVM-RFE in the grid $C_{FGM} = [0.001, 0.01, 0.1, 1, 5, 10]$ and choose $C_{FGM} = 5$ which gives good performance for both FGM and SVM-RFE. This is the same as [18]. For $\text{FGM}_{multi}^{hamming}$, we do the experiments by fixing $C_{FGM_{multi}}$ as $0.1 \times n$ for *URL1* and $1.0 \times n$ for *News20.binary*. The setting for budget parameter $B = [2, 5, 10, 50, 100, 150, 200, 250]$ for *News20.binary*, and $B = [2, 5, 10, 20, 30, 40, 50, 60]$ for *URL1*. The elimination scheme of features for SVM-RFE method can be referred to [18]. For l_1 -SVM, we report the results of different C values so as to obtain different number of selected features.

Figure 1 reports the testing accuracy on different datasets. The testing accuracy is comparable among different methods, but both $\text{FGM}_{multi}^{hamming}$ and FGM can obtain better prediction performances than SVM-RFE in a small number (less than 20) of selected features on both *News20.binary* and *URL1*. These results show that the proposed method with hamming loss can work well on feature selection tasks especially when choosing only a few features. $\text{FGM}_{multi}^{hamming}$ also performs better than l_1 -SVM on *News20.binary* in most range of selected features. This is possibly because l_1 models are more sensitive to noisy or redundant features on *News20.binary* dataset.

Figure 2 shows that our method with the small B will select smaller number of features than the large B . We also observed that most of features selected by the small B also appeared in the subset of features using the large B . This phenomenon can be obviously observed on *News20.binary*. This leads to the conclusion that $\text{FGM}_{multi}^{hamming}$ can select the important features in the given datasets due to the insensitivity of parameter B . However, we notice that not all the features in the selected subset of features with smaller B fall into that of subset of features with the large B , so our method is non-monotonic feature

²www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets

selection. This argument is consistent with the test accuracy in Figure 1. *News20.binary* seems to be monotonic datasets from Figure 2, since $\text{FGM}_{\text{multi}}^{\text{hamming}}$, FGM and SVM-RFE demonstrate similar performance. However, *URL1* is more likely to be non-monotonic, as our method and FGM can do better than SVM-RFE. All the facts imply that the proposed method is comparable with FGM and SVM-RFE, but our method also demonstrates the non-monotonic property for feature selection.

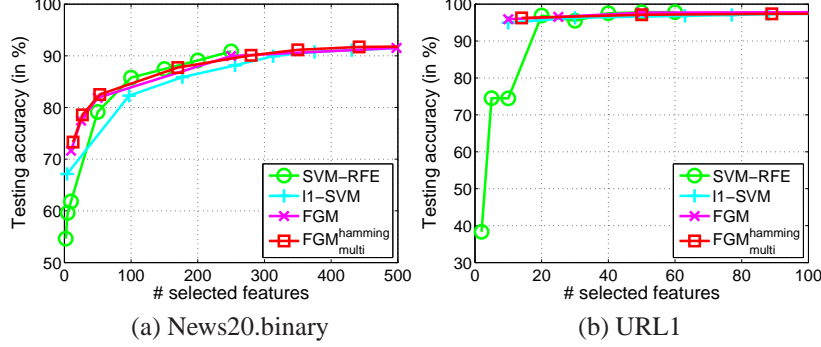


FIGURE 1. Testing accuracy on different datasets

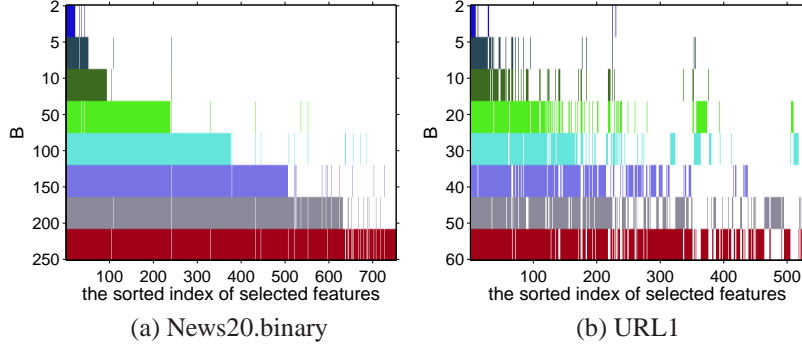
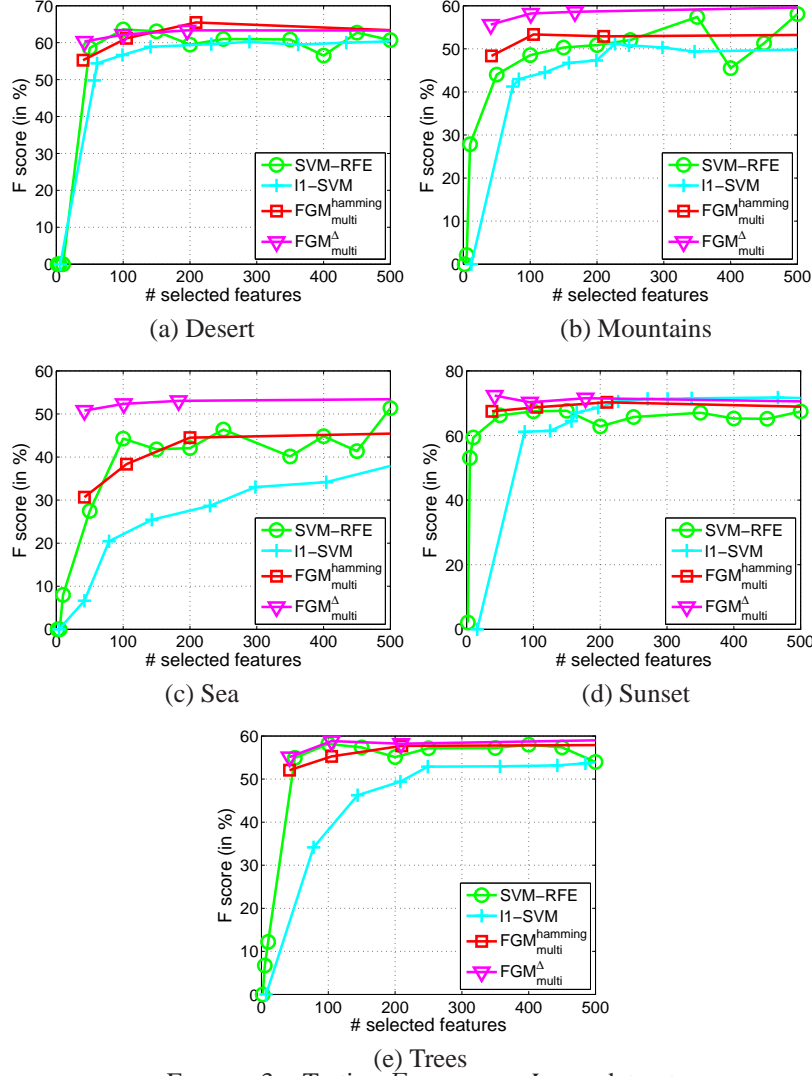


FIGURE 2. The sparsity of features of $\text{FGM}_{\text{multi}}^{\text{hamming}}$ with varying B on different datasets. Each row bar with different color represents the different subset of features selected under current B , where the white region means the features are not selected.

5.2. Feature Selection for Image Retrieval. In this subsection, we demonstrate the specific multivariate performance measures are important to select features for real applications. In particular, we evaluate F_1 measure (commonly used performance measure) for the task of image retrieval. Due to the success of transforming multiple instance learning into a feature selection problem by embedded instance selection, we use the same strategy in Algorithm 4.1 of [4] to construct a dense and high dimensional dataset on a preprocessed image data³. This dataset is used in [27] for multi-instance learning. It contains

³<http://cs.nju.edu.cn/zhouch/zhouch.files/publication/annex/miml-image-data.htm>

FIGURE 3. Testing F_1 scores on *Image* dataset.

five categories and 2,000 images, each image is represented as a bag of nine instances generated by the SBN method [13]. Each image bag is represented by a collection of nine 15-dimensional feature vectors. After that, following [4], the natural scene image retrieval problem turns out to be a feature selection task to select relevant embedded instances for prediction. The *Image* dataset are split randomly with the proportion of 60% for training and 40% for testing (Table 1). Since F_1 -score is used for performance metric, we perform FGM_{multi}^{Δ} for F_1 -score, namely $FGM_{multi}^{F_1}$ as well as other state-of-the-art feature selection methods. As mentioned above, FGM and $FGM_{multi}^{hamming}$ have similar performances, we will not report the results of FGM here. $FGM_{multi}^{hamming}$ and FGM_{multi}^{Δ} use the fixed $C = 10 \times n$. For other methods, we use the previous settings. The testing F_1 values of all methods on each category are reported in Figure 3.

TABLE 2. The macro-average testing performance comparisons among different methods. The quantities in the parentheses represent won/lost of the current method comparing with $\text{FGM}_{multi}^\Delta$. The last column indicates the average number of features is actually used in the current method for a specific measure.

Dataset	method	F_1	$Rec@2p$	$PRBEP$	#selected features
Rcv1	$\text{FGM}_{multi}^\Delta$	42.68	67.81	58.01	690.4/673.7/547.8
	$\text{FGM}_{multi}^{hamming}$	26.81 (0/51)	36.19 (0/50)	31.30 (0/49)	451.6
	$\text{SVM}_{multi}^\Delta$	26.55 (5/46)	63.05 (24/26)	55.48 (15/35)	47,236
Sector	$\text{FGM}_{multi}^\Delta$	92.07	95.77	93.25	787.6/658.9/508.3
	$\text{FGM}_{multi}^{hamming}$	84.99 (12/91)	90.01 (0/71)	85.54 (0/86)	689.2
	$\text{SVM}_{multi}^\Delta$	33.35 (1/104)	95.52 (11/19)	91.24 (11/47)	55,197
News20	$\text{FGM}_{multi}^\Delta$	77.56	91.21	81.46	1,301 / 1,186 / 931
	$\text{FGM}_{multi}^{hamming}$	49.61 (0/20)	66.32 (0/20)	52.14 (0/20)	485.1
	$\text{SVM}_{multi}^\Delta$	55.53 (0/20)	93.08 (16/2)	80.83 (6/11)	62,061

From Figure 3, we observe that $\text{FGM}_{multi}^{F_1}$ and $\text{FGM}_{multi}^{hamming}$ achieve significantly improved performance over l_1 methods in term of F_1 -score especially when choosing less than 100 features. Moreover, SVM-RFE also outperforms l_1 methods on three categories out of five. This verifies that ℓ_1 penalty does not perform as well as ℓ_0 methods like $\text{FGM}_{multi}^{F_1}$ and $\text{FGM}_{multi}^{hamming}$ on dense and high dimensional datasets. It is possibly because ℓ_1 -norm penalty is very sensitive to dense and noisy features. We also observe that $\text{FGM}_{multi}^{F_1}$ performs better than $\text{FGM}_{multi}^{hamming}$ and SVM-RFE on four over five categories. All these facts imply that directly optimizing F_1 measures is useful to boost F_1 performance measure.

5.3. Multivariate Performance Measures for Document Retrieval. In this subsection, we focus on feature selection for different multivariate performance measures on imbalanced text data shown in Table 1. For multiclass classification problems, one vs. rest strategy is used. The comparing model is SVM^{perf} ⁴. Following [8], we use the same notation $\text{SVM}_{multi}^\Delta$ for different multivariate performance measures. The command used for training SVM^{perf} can work for different measures by $-l$ option⁵. In our experiments, we search the C_{perf} in the same range $[2^{-6}, \dots, 2^6]$ as in [8]. We choose the one which demonstrates the best performance of $\text{SVM}_{multi}^\Delta$ to each multivariate performance measure for comparison. $\text{FGM}_{multi}^\Delta$ and $\text{FGM}_{multi}^{hamming}$ fix $C_{FGM_{multi}} = 0.1 \times n$ for *Rcv1* and *News20* except $1.0 \times n$ for *Sector*. For $Rec@k$, we use k as twice the number of positive examples, namely $Rec@2p$. The evaluation for this measure uses the same strategy to label twice the number of positive examples as positive in the test datasets, and then calculate $Rec@2p$.

Table 2 shows the macro-average of the performance over all classes in a collection in which both $\text{FGM}_{multi}^\Delta$ and $\text{FGM}_{multi}^{hamming}$ with only $B = 250$ are listed. The improvement of $\text{FGM}_{multi}^\Delta$ over $\text{FGM}_{multi}^{hamming}$ and $\text{SVM}_{multi}^\Delta$ with respect to different B values are reported in Figure 4. From Table 2, $\text{FGM}_{multi}^\Delta$ is consistently better than $\text{FGM}_{multi}^{hamming}$ on all multivariate performance measures and three multiclass datasets. Similar results can be

⁴ www.cs.cornell.edu/People/tj/svm_light/svm_perf.html

⁵ `svm_perf_learn -c C_{perf} -w 3 -b 0 train_file train_model`

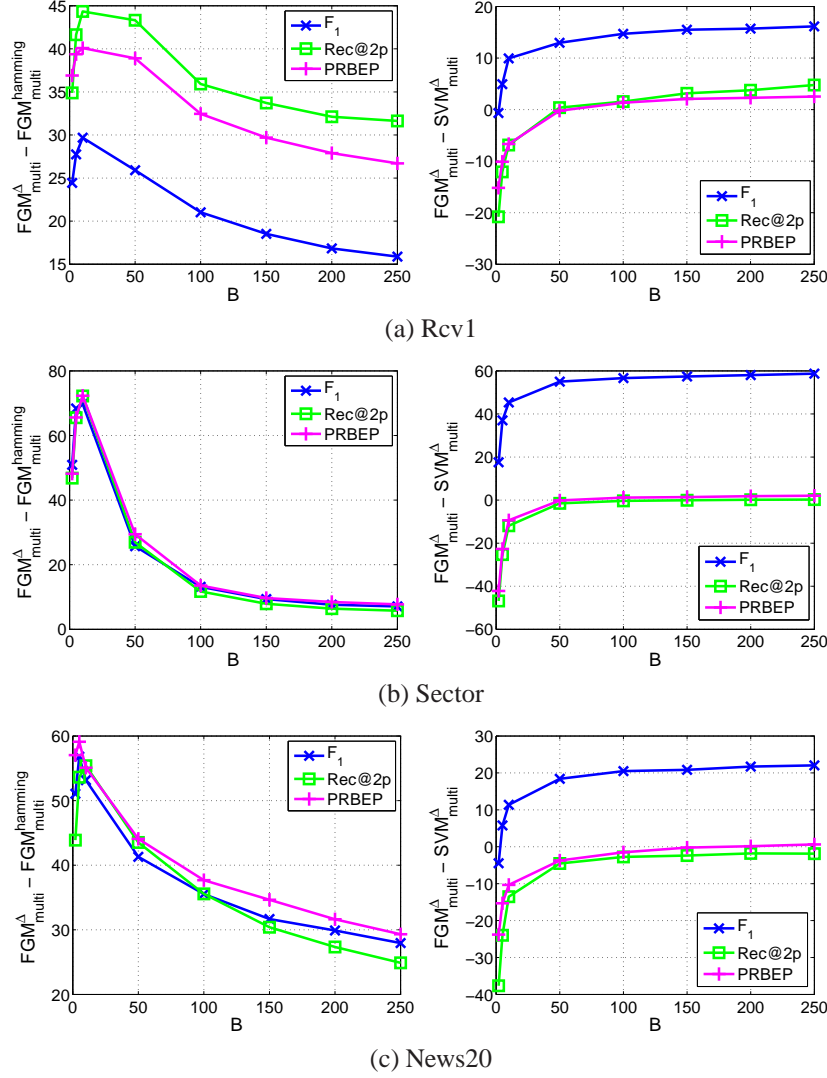


FIGURE 4. The average performance improvement of FGM_{multi}^{Δ} with varying B on different datasets.

obtained comparing with SVM_{multi}^{Δ} , while the only exception is the measure $Rec@2p$ on *News20* where SVM_{multi}^{Δ} is a little better than FGM_{multi}^{Δ} . The largest gains are observed for F_1 score on all three text classification tasks. This implies that a small number of features selected by FGM_{multi}^{Δ} is enough to obtain comparable or even better performances for different measures with SVM_{multi}^{Δ} using all features.

From Figure 4, FGM_{multi}^{Δ} consistently performs better than $FGM_{multi}^{hamming}$ for all of the multivariate performance measures from the figures in the left-hand side. Moreover, the figures in the right-hand side show that the small number of features are good for F_1 measures, but poor for other measures. As the number of features increases, $Rec@2p$ and

$PRBEP$ can approach to the results of SVM_{multi}^{Δ} and all curves become flat. The performance of $PRBEP$ and $Rec@2p$ is relatively stable when sufficient features are selected, but our method can choose very few features for fast prediction. For F_1 measure, our method is consistently better than SVM_{multi}^{Δ} , and the results show significant improvement over all range of B . This improvement may be due to the reduction of noisy or non-informative features. Furthermore, FGM_{multi}^{Δ} can achieve better performance measures than $FGM_{multi}^{hamming}$.

6. CONCLUSION

Learning algorithms need application specific performance measures to evaluate its success. Due to the high dimensionality of the data in many applications, SVM for multivariate performance measures on full set of features may degrade. In this paper, we propose a learning framework to train the SVM model for multivariate performance measures and do feature selection at the same time. To solve this optimization problem, a two-layer cutting plane algorithm was proposed. Experimental results showed that the proposed method is comparable with FGM and SVM-RFE and better than l_1 models on feature selection task, and outperforms SVM for multivariate performance measures on full set of features.

REFERENCES

- [1] E. D. Andersen and A. D. Andersen. *The MOSEK interior point optimizer for linear programming: An implementation of the homogeneous algorithm*. Kluwer Academic Publishers, 2000.
- [2] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *International Conference on Machine Learning*, 2004.
- [3] A. B. Chan, N. Vasconcelos, and G. R. G. Lanckriet. Direct convex relaxations of sparse SVM. In *International Conference on Machine Learning*, 2007.
- [4] Y. Chen, J. Bi, and J. Z. Wang. MILES: Multiple-instance learning via embedded instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1931–1947, 2006.
- [5] G. M. Fung and O. L. Mangasarian. A feature selection newton method for support vector machine classification. *Computational Optimization and Applications*, 28:185–202, 2004.
- [6] I. Guyou, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002.
- [7] J. B. Hiriart-Urruty and C. Lemarechal. *Convex Analysis and Minimization Algorithms*. Springer-Verlag, 1993.
- [8] T. Joachims. A support vector method for multivariate performance measures. In *International Conference on Machine Learning*, 2005.
- [9] J. E. Kelley. The cutting plane algorithm for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.
- [10] S.-J. Kim and S. Boyd. A minimax theorem with applications to machine learning, signal processing and finance. *SIAM Journal on Optimization*, 2008.
- [11] Y. F. Li, I. W. Tsang, J. T. Kwok, and Z.H. Zhou. Tighter and convex maximum margin clustering. In *AI & Statistics*, 2009.
- [12] Z. Liu, F. Jiang, G. Tian, S. Wang, F. Sato, S. J. Meltzer, and M. Tan. Sparse logistic regression with l_p penalty for biomarker identification. *Statistical Applications in Genetics and Molecular Biology*, 6(1), 2007.
- [13] O. Maron and A. L. Ratan. Multiple-instance learning for natural scene classification. In *Proceedings of the 15th International Conference on Machine Learning*, 1998.
- [14] A. Mutapcic and S. Boyd. Cutting-set methods for robust convex optimization with pessimizing oracles. *Optimization Methods & Software*, 24(3):381406, 2009.
- [15] A. Y. Ng. Feature selection, ℓ_1 vs. ℓ_2 regularization, and rotational invariance. In *International Conference on Machine Learning*, 2004.
- [16] A. Rakotomamonjy, F. R. Bach, Y. Grandvalet, and S. Canu. SimpleMKL. *Journal of Machine Learning Research*, 3:1439–1461, 2008.
- [17] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Scholköpfung. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7, 2006.

- [18] M. Tan, L. Wang, and I. W. Tsang. Learning sparse SVM for feature selection on very high dimensional datasets. In *International Conference on Machine Learning*, 2010.
- [19] C. H. Teo, S.V.N. Vishwanathan, A. Smola, and Quoc V. Le. Bundle methods for regularized risk minimization. *Journal of Machine Learning Research* 11, pages 311–365, 2010.
- [20] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altum. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.
- [21] J. Weston, A. Elisseeff, and B. Scholköpf. Use of the zero-norm with linear models and kernel methods. *Journal of Machine Learning Research*, 3:1439–1461, 2003.
- [22] Z. Xu, R. Jin, I. King, and M. R. Lyu. An extended level method for efficient multiple kernel learning. In *Advances in Neural Information Processing Systems*, 2008.
- [23] Z. Xu, R. Jin, J. Ye, Michael R. Lyu, and I. King. Non-monotonic feature selection. In *International Conference on Machine Learning*, 2009.
- [24] G.-X. Yuan, K.-W. Chang, C.-J. Hsieh, and C.-J. Lin. A comparison of optimization methods and software for large-scale l_1 -regularized linear classification. *J. Mach. Learn. Res.*, 11:3183–3234, 2010.
- [25] Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *Proceedings of the Conference on Research and Development in Information Retrieval (SIGIR)*, 2007.
- [26] T. Zhang. Analysis of multi-stage convex relaxation for sparse regularization. *Journal of Machine Learning Research*, 11:1081–1107, Mar 2010.
- [27] Z.-H. Zhou and M.-L. Zhang. Multi-instance multi-label learning with application to scene classification. In *Advances in Neural Information Processing Systems 19 (NIPS'06)*, 2007.
- [28] J. Zhu, S. Rossett, T. Hastie, and R. Tibshirani. 1-norm support vector machine. In *Advances in Neural Information Processing Systems 20*, 2003.